UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/662,034	09/12/2003	Paul J. Mantey	200309970-1	9298

22879 7590 04/04/2006

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
SPITTLE, MATTHEW D	

ART UNIT	PAPER NUMBER
2111	

DATE MAILED: 04/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/662,034	MANTEY ET AL.
	Examiner	Art Unit
	Matthew D. Spittle	2111

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 March 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13,31,32 and 42-44* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *14,16-18,20,21,28-30,33,34,38 and 40* is/are allowed.

6)☒ Claim(s) *1-13,31,32 and 42-44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

Applicant's arguments filed 3/3/2006 have been fully considered but they are not persuasive.

With regard to claims 31 and 32, Applicant has asserted that the Office Action ignores an express limitation of claim 31, namely that the checksum generation messages generates a message checksum for a message *while the message is being received by the bus controller over the communications bus.* Examiner wishes to clarify the rejection by pointing to column 11, lines 14 – 24 of Yoshida, which state "the data transfer is started, and the data check code generation circuits 23 and 73 of the data processing unit and the external recording unit 10 start the calculation." Examiner interprets this passage as meaning that the data check code generation (checksum) is generated (calculated) while the message is being received over the bus.

In response to Applicant's argument that there is no motivation to combine, Examiner maintains that error-free data is always desirable in a digital system to have correct and reliable operation. Additionally, Yoshida provides motivation for using his/her checksum means in column 12, lines 19 – 23. Here Yoshida explains that his/her invention provides correctness of the reception data at the time of the data transfer, versus after the time of the data transfer for example. This would indicate to one of ordinary skill in this art that Yoshida's invention provides correctness of data with

little or no detriment to overall performance since the correctness verification is performed in parallel with the data transfer.

Thus, examiner cannot allow claims 31 and 32.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Amended claims 1, 12, 42, 43, and 44 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claims 1, 12, 42, 43, and 44 recite a first-in first-out (FIFO) buffer coupled to the send machine and coupled between the host processor and the bus controller but not between the send machine and the bus controller. Figure 2A of the applicant's disclosure clearly shows a first FIFO (214) coupled between a send machine (210) and a bus controller (208) by way of bus (218) and bus (216). Similarly, Figure 2A clearly shows a second FIFO (226) coupled between a receive machine (222) and a bus controller (208) by way of bus (230) and bus (228).

Claim Objections

Claim 11 recites the limitation "the byte timer" in lines 1 -2. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1 – 4, 14, 16, 17, 18, 20, and 21 are rejected under 35 U.S.C. 102(b) as being anticipated by Liu et al.

With regard to claim 1, Liu et al. describe a computer system comprising:

A communications bus implemented in accordance with an Inter-IC bus specification (Figure 1, 2, item 107);

A bus controller coupled to the communications bus (where a bus controller may be interpreted as a system interface processor; column 6, lines 19 - 23);

A send machine (Figure 2, items 207) coupled between a host processor (Figure 1, item 101) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 7, lines 16 – 25);

A first first-in first-out (FIFO) buffer coupled to the send machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a request queue; Figure 2, item 203; column 7, lines 16 – 21).

With regard to claim 2, Liu et al. describe the computer system of claim 1, wherein the first FIFO buffer comprises means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 7, lines 25 – 26; Figure 2, items 203, 205 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 10, lines 24 – 34 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

With regard to claim 3, Liu et al. describe the computer system of claim 1, wherein:

The first FIFO buffer comprises means for receiving a plurality of bytes from the host processor ((column 7, lines 25 – 26; Figure 2, items 203, 205 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes);

The send machine comprises means for transmitting the plurality of bytes over the communications bus without interrupt the host processor (column 10, lines 24 – 34 tell of how a messages is placed into the request buffer and processed by the interface,

but note that the interrupt is not generated until a response is received from the I2C device).

With regard to claim 4, Liu et al. describe the computer system of claim 1, further comprising:

A receive machine coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 7, lines 16 – 25);

A second FIFO buffer coupled to the receive machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a response queue; Figure 2, item 205; column 7, lines 16 – 21).

* * *

Claims 12, 28, and 43 are rejected under 35 U.S.C. 102(b) as being anticipated by Johnson et al.

With regard to claim 12, Johnson et al. teach a computer system comprising:

A communications bus implemented in accordance with an Inter-IC bus specification (Figure 4, 7, item 310; column 7, lines 10 - 12);

A bus controller coupled to the communications bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65);

A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the send machine comprising means for transmitting the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

A first first-in first-out (FIFO) buffer coupled to the send machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the receive machine comprising means for receiving the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the

response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

A second FIFO buffer coupled to the receive machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14), the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

With regard to claim 28, Johnson et al. describe a method for transmitting a message comprising a plurality of bytes from a source device having a first host processor to a destination device having a second host processor

With regard to claim 43, Johnson et al. teach a computer system comprising:

A communications bus (Figure 4, 7, item 310);

A bus controller coupled to the communications bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65);

A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the send machine comprising means for transmitting the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

A first first-in first-out (FIFO) buffer coupled to the send machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the receive machine comprising means for

receiving the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

A second FIFO buffer coupled to the receive machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14), the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Liu et al. in view of Yoshida.

Liu et al. fail to teach wherein the receive machine comprises checksum generation means for generating a message checksum for a message while the message is being received by the bus controller over the communications bus.

Yoshida teaches a checksum generation means for generating a message checksum for a message while the message is being received by the bus controller over the communications bus (where checksum generation means may be interpreted as data check code generation circuits; column 11, lines 14 – 24).

It would have been obvious to one or ordinary skill in this art at the time of invention by applicant to combine the checksum generation means of Yoshida with the system of Liu et al. in order to provide for a means of verifying the data transmitted

across the communications bus. This would have been obvious since error-free data is critical to the correct operation of a digital system.

* * *

Claims 6 – 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Liu et al. in view of Feeney et al.

With regard to claim 6, Liu et al. describe the computer system of claim 1, further comprising:

Means for receiving a message from the host processor (Figure 2, items 203, 207; Figure 4A, item 420);

Means for attempting to send the message over the communications bus to a target device (Figure 4A, item 422);

Liu et al. fail to teach:

Means for determining whether the message was received without errors by the target device;

Retry means for attempting again to send the message over the communication bus to the target device if it is determined that the message was not received without errors by the target;

Feeney et al. teach means for determining whether the message was received without errors through the use of FIFO status registers (Figures 13, 14; column 18, line 4 – column 20, line 12), and retry means for attempting again to send the message over

the communication bus to the target if it is determined that the message was not received without errors by the target (column 16, lines 36 – 49 describe retrying messages that failed to send).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the means of retrying failed messages as taught by Feeney et al. into the computer system of Liu et al for the purpose of ensuring the delivery of messages on the communication bus.

With regard to claim 7, Feeney et al. teach the additional limitation wherein the retry means comprises means for attempting again to send the message over the communications bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe retrying a message without involving the processor).

With regard to claim 8, Feeney et al. teach the additional limitation wherein the retry means comprises means for attempting again to send the message over the communications bus to the target device without obtaining the message again from the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe storing the message in a FIFO in order to allow the processor to move onto other tasks).

* * *

Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Liu et al. in view of Cao et al.

Liu et al. fail to teach a busfree count means for storing a busfree count associated with the computer system, a busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the communications bus after the communications bus becomes available for use, and a fair arbitration block coupled between the host processor and the bus controller, the fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal.

Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the communications bus after the communications bus becomes available for use (where a busfree timer may be interpreted as a "quiet slot" counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Liu et al. for the purpose of providing arbitration amongst devices on the communications bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwith (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

* * *

Claims 10 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Liu et al. in view of Webb et al.

With regard to claim 10, Liu et al. fail to teach a byte timer coupled between the bus controller and the host processor.

Webb et al. teach a byte timer (where a byte timer may be interpreted as a no-response timer; column 13, lines 49 - 60).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system of Liu et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the system from wasting time sending messages to processors/terminals that are not responsive (column 14, lines 11 – 19).

With regard to claim 11, Webb et al. teach the additional limitation wherein the byte timer (interpreted as a no-response timer) comprises means for determining whether the host processor has failed and means for generating a signal indicating whether the host processor has failed (where a host processor may be interpreted as a terminal; where generating a signal indicating the processor has failed may be interpreted as marking a terminal as being "offline or "down"; column 13, line 49 – column 14, line 30).

* * *

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Feeney et al., in view of Cao et al., and further in view of Webb et al.

Johnson et al. teach a computer system of claim 12 further comprising:

Means for receiving a message from the host processor (Figure 7, items 516, 707; column 12, lines 26 – 32);

Means for attempting to send the message over the communications bus to a target device (column 15, lines 15 – 36 give an example of how a message is sent over the communications bus to a target device (microcontroller)).

Means for determining whether the message was received without errors by the target device (column 15, lines 62 – 64).

Johnson et al. fail to describe a retry means, a busfree count means, a busfree count timer, a fair arbitration block, and a byte timer.

Feeney et al. teach retry means for attempting again to send the message over the communications bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe retrying messages that failed to send; column 16, lines 36 – 49 describe retrying a message without involving the processor).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the means of retrying failed messages as taught by Feeney et al. into the computer system of Liu et al for the purpose of ensuring the delivery of messages on the communication bus.

Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the communications bus after the communications bus becomes available for use (where a busfree timer may be interpreted as a "quiet slot" counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Liu et al. for the purpose of providing arbitration amongst devices on the communications bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwith (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

Webb et al. teach a byte timer comprising means for determining whether the host processor has failed and means for generating a signal indicating whether the host processor has failed (where a byte timer may be interpreted as a no-response timer; column 13, lines 49 – 60; where a host processor may be interpreted as a terminal; where generating a signal indicating the processor has failed may be interpreted as marking a terminal as being “offline or “down”; column 13, line 49 – column 14, line 30).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system of Liu et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the system from wasting time sending messages to processors/terminals that are not responsive (column 14, lines 11 – 19).

* * *

Claims 31 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Liu et al. in view of Yoshida.

Liu et al. teach a computer system comprising:

A communications bus implemented in accordance with an Inter-IC bus specification (Figure 1, 2, item 107);

A bus controller coupled to the communications bus (where a bus controller may be interpreted as a system interface processor; column 6, lines 19 – 23);

A receive machine coupled to the communications bus (where a receive machine may be interpreted as a message data register (MDR); column 7, lines 16 – 25);

A host processor coupled to the receive machine (Figure 1, item 101);

Liu et al. fail to teach wherein the receive machine comprises checksum generation means for generating a message checksum for a message while the message is being received by the bus controller over the communications bus.

Yoshida teaches a checksum generation means for generating a message checksum for a message while the message is being received by the bus controller over the communications bus (where checksum generation means may be interpreted as data check code generation circuits; column 11, lines 14 – 24).

It would have been obvious to one or ordinary skill in this art at the time of invention by applicant to combine the checksum generation means of Yoshida with the system of Liu et al. in order to provide for a means of verifying the data transmitted across the communications bus. This would have been obvious since error-free data is critical to the correct operation of a digital system.

With regard to claim 32, Yoshida teaches the additional limitation wherein the checksum generation means comprises means for generating the message checksum without interrupting the host processor (column 11, lines 14 – 24 describe the data check code generation circuits (Figure 21, item 23) generating the checksum, which examiner identifies as separate hardware from the host processor (interpreted as a data processor in this reference).

* * *

Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Feeney et al., in view of Cao et al., and further in view of Webb et al.

Johnson et al. teach a computer system comprising:

A communications bus (Figure 4, 7, item 310);

A bus controller coupled to the communications bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65);

A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the send machine comprising means for transmitting the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface,

but note that the interrupt may not be generated until a response is received from the I2C device).

A first first-in first-out (FIFO) buffer coupled to the send machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the receive machine comprising means for receiving the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

A second FIFO buffer coupled to the receive machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14), the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

Means for receiving a message from the host processor (Figure 7, items 516, 707; column 12, lines 26 – 32);

Means for attempting to send the message over the communications bus to a target device (column 15, lines 15 – 36 give an example of how a message is sent over the communications bus to a target device (microcontroller)).

Means for determining whether the message was received without errors by the target device (column 15, lines 62 – 64).

Johnson et al. fail to describe a retry means, a busfree count means, a busfree count timer, a fair arbitration block, and a byte timer.

Feeney et al. teach retry means for attempting again to send the message over the communications bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device

(column 16, lines 36 – 49 describe retrying messages that failed to send; column 16, lines 36 – 49 describe retrying a message without involving the processor).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the means of retrying failed messages as taught by Feeney et al. into the computer system of Liu et al for the purpose of ensuring the delivery of messages on the communication bus.

Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the communications bus after the communications bus becomes available for use (where a busfree timer may be interpreted as a "quiet slot" counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Liu et al. for the purpose of providing arbitration amongst devices on the communications bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwith (column 10,

lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

Webb et al. teach a byte timer comprising means for determining whether the host processor has failed and means for generating a signal indicating whether the host processor has failed (where a byte timer may be interpreted as a no-response timer; column 13, lines 49 – 60; where a host processor may be interpreted as a terminal; where generating a signal indicating the processor has failed may be interpreted as marking a terminal as being "offline or "down"; column 13, line 49 – column 14, line 30).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system of Liu et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the system from wasting time sending messages to processors/terminals that are not responsive (column 14, lines 11 – 19).

* * *

Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Cao et al.

Johnson et al. teach a device for use in a computer system including a communications bus and a bus controller coupled to the communications bus, the device comprising:

A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the send machine comprising means for transmitting the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

A first first-in first-out (FIFO) buffer coupled to the send machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the receive machine comprising means for receiving the plurality of bytes over the communications bus without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the

response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

A second FIFO buffer coupled to the receive machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14), the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

Johnson et al. fail to describe a busfree count storage means, a busfree count timer, a fair arbitration block, and a byte timer.

Cao et al. teach:

A busfree count storage means for storing a busfree count (where a busfree count may be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the communications bus

after the communications bus becomes available for use (where a busfree timer may be interpreted as a "quiet slot" counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Liu et al. for the purpose of providing arbitration amongst devices on the communications bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwith (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

Allowable Subject Matter

Claims 14, 16 – 18, 20, 21, 28 – 30, 33, 34, 38, 40, and 41 are allowed.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Matthew D. Spittle whose telephone number is (571) 272-2467. The examiner can normally be reached on Monday - Friday, 8 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Matthew Spittle

MDS

John R. Cottingham

JOHN R. COTTINGHAM
PRIMARY EXAMINER